Chapter 10

Digital-to-Analogue Converter

The digital-to-analogue converter (DAC) is the last element of a signal chain and the interface between the microcontroller and the real analogue world. The DAC converts digital signals to analogue electrical signals. This chapter presents the DAC architecture and operation and then develops an exercise to illustrate the concepts.

Topic **Page**

10.1 Introduction to Digital to Analogue Conversion

Nowadays, digital devices are used worldwide and have become integral parts of our daily lives. However, real world signals are analogue in nature. Thus, the final part of an analogue signal chain requires a digital to analogue conversion to be able to drive devices. Therefore, it is necessary to utilize an electronic circuit called a digital to analogue converter (DAC) to convert a digital representation of a quantity into an analogue electrical value.

The input digital data of a DAC, along with a known reference voltage, results in an output analogue voltage or current. The DAC provides a continuous output signal, mathematically often treated as discrete Dirac pulses into a zero-order hold and consisting of a series of fixed steps to form a discrete-time output signal instead of a continuous one. Filtering this discrete-time output signal can provide an approximate continuous-time output, as well as increasing the DAC's resolution. Since resolution is determined by the number of discrete levels, increasing the resolution for a fixed voltage or current range consequently reduces the step size and thus reduces the quantization error.

The output of an ideal DAC is a sequence of impulses that are filtered to construct the continuous-time analogue signal. Ideally the DAC should reproduce a sampled signal precisely up to the Nyquist frequency. However, in practice the analogue signal is not a precise reconstruction, since the filter has finite phase delay and there will be quantization errors.

Instead of impulses, the digital data sequence is converted into an analogue voltage at a uniform sampling rate. During a single time period, the clock signal latches the digital data sequence and at the DAC output, an analogue voltage is held.

The most common types of digital-to-analogue converters are:

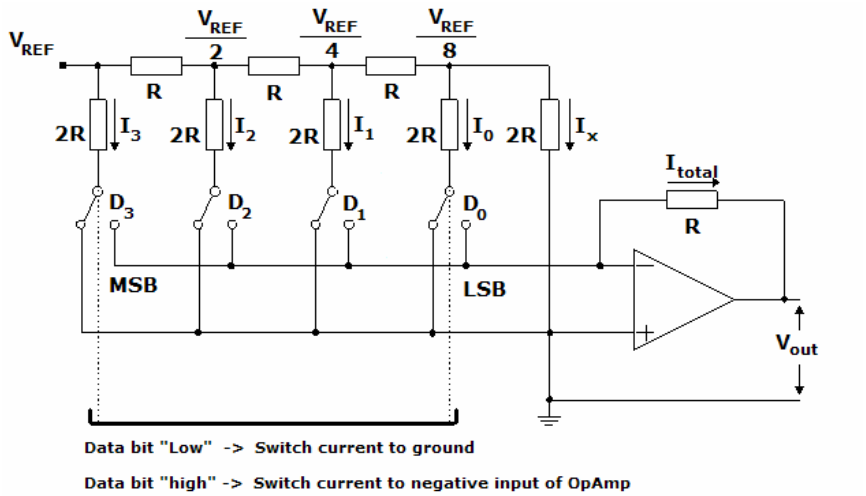
- Binary Weighted DAC: Contains one resistor (or current source) for each bit of the DAC connected to a summing point to the reference voltage value. The DAC's inputs control a set of switches that divide the reference voltage through matched series resistors. The matching requirements for the integrated resistors (or current sources) incorporated in this DAC represent a major design challenge for the binary weighted DAC architecture.
- R/2R Ladder DAC: Uses a repeating cascaded structure of resistor values R and 2R to create a binary weighted DAC. The DAC12 module included in the MSP430 uses this architecture. The MSB's weighted binary digit position value is equal to one-half the value of the full code value, and that the next least significant bit is one-half the MSB's digit position value. The R/2R ladder divides down a positive reference voltage by switching individual resistors between a positive reference voltage, $V_{REF,}$ and the analogue ground, generating a current. The equivalent resistance between V_{REF} and ground is R. An operational amplifier converts this current

to voltage and provides a low impedance output (see *Figure 10-1* for a 4-bit example DAC);

- Interpolating DAC: Uses a pulse density conversion technique that was given in the Sigma Delta ADC section (Chapter 9 –Data Acquisition);
- Pulse Width Modulator: Assumes that a stable voltage or current is switched into a low pass analogue filter for a time period representing the digital input code;
- Thermometer coded DAC: Uses one resistor or current source segment for each possible value of the DAC output. Due to the number of segments (2^n-1, where *n* is the number of bits) needed, this type of DAC is very expensive despite its high precision and conversion speed;
- Hybrid DAC: Uses a combination of the above techniques in a single converter;

Segmented DAC: Consists of a thermometer coded approach for the most significant bits and a binary weighted approach for the least significant bits.

Figure 10-1. R/2R 4 bit DAC architecture.



The characteristic parameters of a DAC are similar to those of an ADC. (See *Chapter 9*). The most important parameters are:

- Resolution (*n*): Number of possible output levels, 2^n, the DAC can reproduce, given by the number of bits, *n*. Actual resolution for a DAC, however, is given by the Effective Number Of Bits (ENOB);
- Integral NonLinearity (INL): Deviation of a DAC's transfer function from a straight line;

- Differential NonLinearity (DNL): Difference between an actual step height and the ideal value of 1 LSB (if DNL <1 LSB, the DAC is monotonic, i.e., no data is lost);

- Offset error: Analogue output voltage value when the digital input is zero;

- Gain error: Difference between the ideal maximum output voltage and the actual maximum value of the transfer function, after subtracting the offset error;

- Monotonicity: Ability of a DAC's analogue output to increase with an increase in digital code, or the converse;

- Total Harmonic Distortion (THD): Distortion and noise introduced to the signal by the DAC. This is a very important DAC characteristic for dynamic and small signal DAC applications;

- Dynamic range: Difference between the largest and smallest signals the DAC can reproduce.

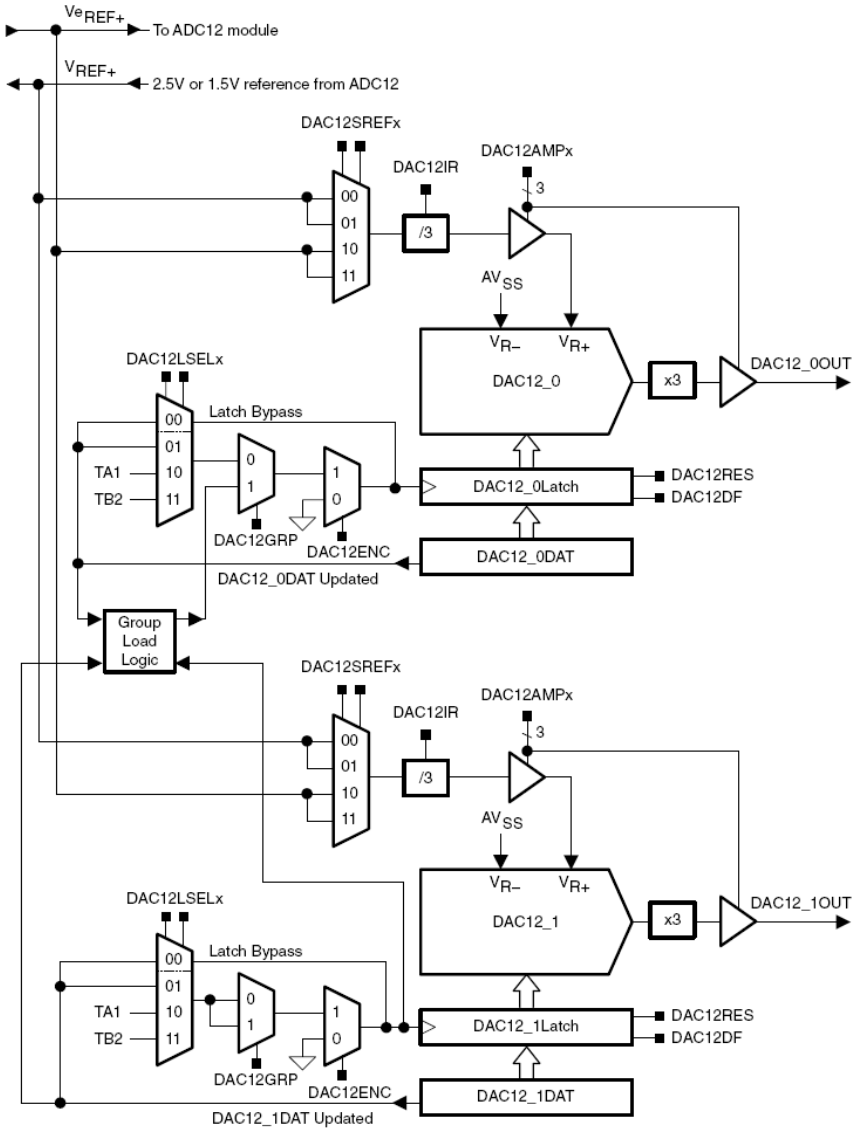
10.2 DAC operation

The 12-bit DAC12 module is a voltage output DAC. All the hardware development tools have this module. The MSP430FG4618 device populated in the Experimenter's board has two DAC12 modules. This allows the board to group both DACs together for operations with synchronous updates.

DAC12 features:

- 12-bit monotonic output;
- 8-bit or 12-bit voltage output resolution;
- Programmable settling time *vs.* power consumption;
- Internal or external reference selection;
- Straight binary or Two's complement data format;
- Self-calibration option for offset correction;
- Synchronized update capability for multiple DAC12s;
- Direct Memory Access (DMA) enabled.

Figure 10-2. DAC12 block diagram.



DAC12 core

The dynamic range of the DAC12 is controlled by the DAC's resolution (8-bit or 12-bit configurable with DAC12RES bit), and full-scale output (1xV_{REF} or 3xV_{REF} configurable with DAC12IR bit). There are two configurable output formats: straight binary or Two's complement (using the DAC12DF bit).

The output voltage for straight binary data format is given in *Table 10–1* and shows the possible resolutions.

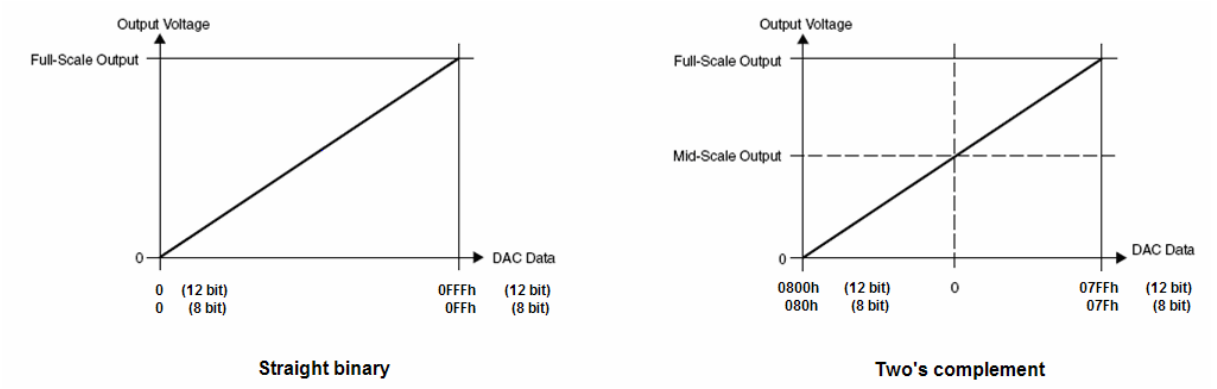
Table 10-1. Output voltage.

Resolution	DAC12RES	DAC12IR	Output voltage
12-bit	0	0	$V_{OUT} = V_{REF} \times 3 \times \frac{DAC12\_xDAT}{4096}$
12-bit	0	1	$V_{OUT} = V_{REF} \times \frac{DAC12\_xDAT}{4096}$
8-bit	1	0	$V_{OUT} = V_{REF} \times 3 \times \frac{DAC12\_xDAT}{256}$
8-bit	1	1	$V_{OUT} = V_{REF} \times \frac{DAC12\_xDAT}{256}$

DAC12_xDAT Data Format

The data format modifies the full-scale output voltage as shown in *Figure 10-3*.

Figure 10-3. Output voltage vs. Data format and Resolution.

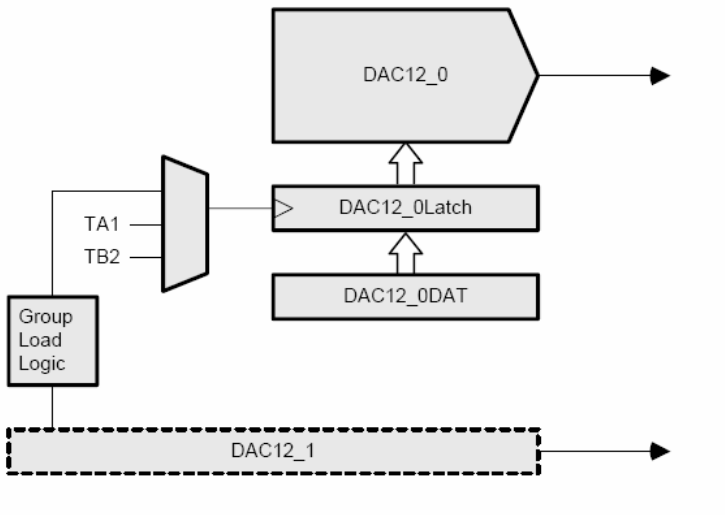


Updating the DAC12 voltage output

The DAC12 voltage output (DAC12_xDAT register) is configurable with the DAC12LSELx bits. The update can be (see *Figure 10-4*):

- DAC12LSELx = 0: Immediate:
  - DAC12 output updates immediately when a new DAC12 data value is written to the DAC12_xDAT register.
- DAC12LSELx = 1: Grouped:
  - DAC12 data value is latched and applied to the DAC12 core after a new data value is written to the DAC12_xDAT register.
- DAC12LSELx = 2: rising edge from the Timer_A CCR1 output;
- DAC12LSELx = 3: rising edge from the Timer_B CCR2 output.

Figure 10-4. Output voltage update options.



Timer Triggered DAC

The timer trigger allows for fewer samples and more bandwidth. Although with more samples the uncertainty is less significant, the analogue signal will require more processing overhead and a reduced bandwidth.

DAC12 Interrupts

The DAC12 interrupt vector is shared with the DMA controller.

This structure provides:

- Increased system flexibility;
- No code execution required (see *Figure 10-5*);
- Lower power (see *Figure 10-6*);
- Higher efficiency;

Figure 10-5. DAC12 interrupt vector shared with the DMA controller – increased peripheral performance.

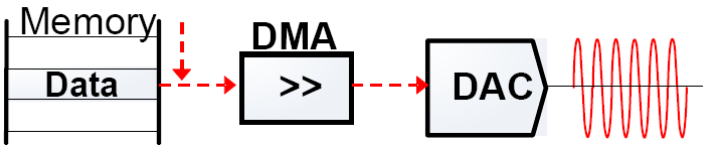
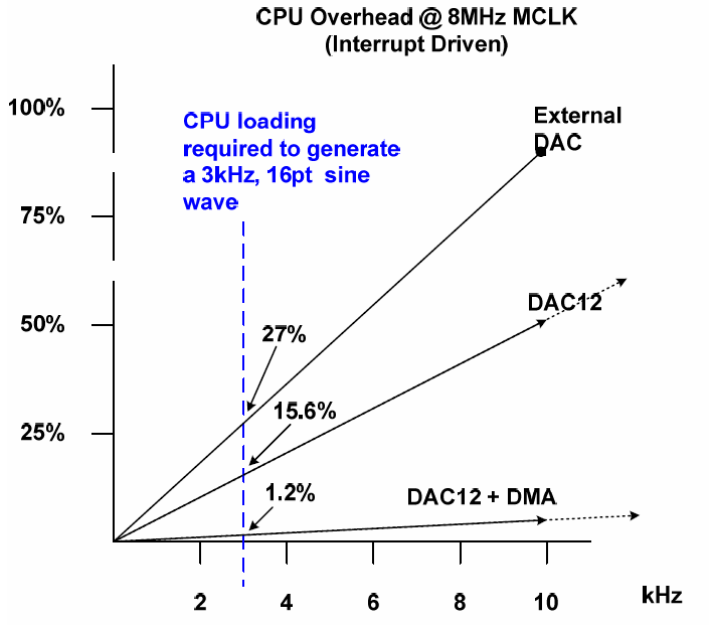


Figure 10-6. CPU overhead using different DAC12 structures.



To determine the source of an interrupt:

- DAC12IFG = 1 (DAC12 is ready for new data):
  - DAC12LSELx > 0;
  - DAC12 data value is latched from the DAC12_xDAT register;
  - DAC12IE = 1 and GIE =1, it generates an interrupt request.
- DAC12IFG = 0:
  - DAC12LSELx = 0;
  - It is only reset by software.

10.3 DAC12 registers

The DAC12 registers used by the MSP430FG4618:

DAC12_xCTL, DAC12 Control Register

15	14 13	12	11 10	9	8
DAC12OPS	DAC12SREFx	DAC12RES	DAC12LSELx	DAC12CALON	DAC12IR

7 6 5	4	3	2	1	0
DAC12AMPx	DAC12DF	DAC12IE	DAC12IFG	DAC12ENC	DAC12GRP

Bit		Description
15	DAC12OPS	DAC12 output: DAC12OPS = 0 ⇒ DAC12_0 on P6.6, DAC12_1 on P6.7 DAC12OPS = 1 ⇒ DAC12_0 on V_{eREF+}, DAC12_1 on P5.1
14-13	DAC12SREFx	DAC12 reference voltage: DAC12REF1 DAC12REF0 = 00 ⇒ V_{REF+} DAC12REF1 DAC12REF0 = 01 ⇒ V_{REF+} DAC12REF1 DAC12REF0 = 10 ⇒ Ve_{REF+} DAC12REF1 DAC12REF0 = 11 ⇒ Ve_{REF+}
12	DAC12RES	DAC12 resolution: DAC12RES = 0 ⇒ 12 bit resolution DAC12RES = 1 ⇒ 8 bit resolution
11-10	DAC12LSELx	DAC12 load: DAC12LSEL1 DAC12LSEL0 = 00 ⇒DAC12_xDAT written DAC12LSEL1 DAC12LSEL0 = 01 ⇒all grouped DAC12_xDAT written DAC12LSEL1 DAC12LSEL0 = 10 ⇒Rising edge of Timer_A.OUT1 (TA1) DAC12LSEL1 DAC12LSEL0 = 11 ⇒Rising edge of Timer_B.OUT2 (TB2)
9	DAC12CALON	DAC12 calibration initialized or in progress when DAC12CALON = 1
8	DAC12IR	DAC12 input range: DAC12IR = 0 ⇒ DAC12 full-scale output: 3x reference voltage DAC12IR = 1 ⇒ DAC12 full-scale output: 1x reference voltage

Bit		Description		
7-5	DAC12AMPx	DAC12 amplifier setting:		
			Input buffer:	Output buffer:
		AMP2 AMP1 AMP0 = 000 ⇒	Off	DAC12 off (high Z)
		AMP2 AMP1 AMP0 = 001 ⇒	Off	DAC12 off (0 V)
		AMP2 AMP1 AMP0 = 010 ⇒	Low f / I	Low f / I
		AMP2 AMP1 AMP0 = 011 ⇒	Low f / I	Medium f / I
		AMP2 AMP1 AMP0 = 100 ⇒	Low f / I	High f / I
		AMP2 AMP1 AMP0 = 101 ⇒	Medium f / I	Medium f / I
		AMP2 AMP1 AMP0 = 110 ⇒	Medium f / I	High f / I
		AMP2 AMP1 AMP0 = 111 ⇒	High f / I	High f / I
		f: frequency (speed) I: current		
4	DAC12DF	DAC12 data format: DAC12DF = 0 ⇒ Straight binary DAC12DF = 1 ⇒ Two's complement		
3	DAC12IE	DAC12 interrupt enable when DAC12IE = 1		
2	DAC12IFG	DAC12 Interrupt flag DAC12IFG = 1 when interrupt pending		
1	DAC12ENC	DAC12 enable when DAC12ENC = 1 and DAC12LSELx>0.		
0	DAC12GRP	Groups DAC12_x with the next higher DAC12_x when DAC12GRP = 1 (exception for DAC12_1)		

DAC12_xDAT, DAC12 Data Register

The 16-bit DAC12_xDAT register always keeps the four most-significant bits (bits 15 – 12) as zeros. The twelve least-significant bits will store the DAC12 data (bits 11 – 0). The DAC12 data value is right justified, but the MSB depends on the resolution (8-bit: Bit 7; 12-bit: Bit 11) and data format (Straight binary: MSB is data; Two's complement: MSB is sign).

10.4 Laboratory 6: Voltage ramp generator

Project files

- C source files: **Chapter 10 > Lab6 > Lab6_student.c**
- Solution file: **Chapter 10 > Lab6 > Lab6_solution.c**

Overview

This laboratory implements a voltage ramp generator. The DAC module reference is obtained from the ADC module. The DAC is configured with 12-bit resolution, in straight binary format. The output of the DAC value is updated once every 1 msec by an interrupt service routine (ISR) generated by Timer_A. The push buttons SW1 and SW2 are used to manually modify the output of the DAC value. When the microcontroller is not performing any task, it enters low power mode.

A. Resources

The DAC12_0 module uses V_{REF+} as reference voltage. It is therefore necessary to activate this reference voltage in the ADC12 module.

The DAC12_0 is connected to Port P6.6 on the Header 8 pin 7. Connect the oscilloscope probe to this port pin.

The output of the DAC is updated whenever Timer_A generates an interrupt. This peripheral is configured to generate an interrupt with a 1 msec time period.

After refreshing the output of the DAC, the system returns to low power mode LPM3.

The push buttons SW1 and SW2 allow the output of the DAC value to be changed manually.

The resources used by the application are:

- Timer_A;
- DAC12;
- I/O ports;
- FLL+;
- Interrupts.

B.Organization of application software

The application starts by stopping the Watchdog Timer.

Then, the ADC12's reference voltage is activated and set to 2.5 V. A delay is used to allow the reference voltage to settle. During this time period, the device enters low power mode LPM0. The delay period, which is controlled by Timer_A, enables an interrupt when it completes. The interrupt wakes the device and proceeds with the execution of the application.

Timer_A is reconfigured to generate an interrupt once every 1 msec. This interrupt service routine (ISR) updates the output of the DAC.

Ports P1.0 and P1.1 are connected to buttons SW1 and SW2. The ports are configured as inputs with interrupt capability, such that the ISR can decode which button is pushed. If the interrupt source is due to button SW1, then the output of the DAC is increased. If the interrupt source is due to button SW2, then the output of the DAC is decreased.

C. System configuration

❑ Reference voltage selection

The DAC12_0 uses the signal V_{REF+} as reference voltage. What is the value to write to the configuration register in order to obtain the internally available reference?

```
ADC12CTL0 = ____________________;
```

❑ DAC12 configuration

The DAC12_0 is configured with 12-bit resolution. The output is updated immediately when a new DAC12 data value is written in straight binary data format to the DAC12_0DAT register.

The full-scale output must be equal to the V_{REF+} 2.5 V internal reference voltage. Choose a compromise solution between the settling time and current consumption, by selecting a medium frequency and current for both input and output buffers. Configure the following register in order to meet these specifications:

```
DAC12_0CTL = ____________________;
```

❑ Timer_A configuration

Configure Timer_A register to enable an interrupt once every 1 msec. Use the ACLK clock signal as the clock source. This timer is configured in count up mode in order to count until the TAR value reaches the TACCR0 value.

```
TACCTL0 = _____________________;
TACCR0 = _____________________;
TACTL = _____________________;
```

- **I/O Ports configuration**

Port P1 uses the bits P1.0 and P1.2 to activate the ISR whenever the push buttons SW1 and SW2 are activated (low-to-high transition).

```
P1SEL = _____________________;
P1DIR = _____________________;
P1IFG = _____________________;
P1IE = _____________________;
```

DAC12_0 is connected to P6.6. Configure P6 as a special function output:

```
P6SEL = _____________________;
P6DIR = _____________________;
```

D. Analysis of operation

- **Observe the analogue signal using an oscilloscope**

After compiling the project and starting the debug session, monitor the operation of the application using an oscilloscope probe connected to pin 7 of Header 8 (P6.6).

- **Measure the current drawn**

Assign different values to the bits set in DAC12AMP0. Suspend the execution of the application then directly change the registers. Do not forget that this change requires suspending the operation of the DAC12 by disabling the bit DAC12ENC. Afterwards, this bit must be enabled.

Please note the special cases relating to:

- DAC12 off;
- High impedance output and DAC12 off;
- Output: 0 V.

MSP-EXP430FG4618 **SOLUTION**

Using the MSP-EXP430FG4618 Development Tool and the MSP430FG4618 device, implement a ramp generator.

- FLL+ configuration:

```
FLL_CTL0 |= DCOPLUS | XCAP18PF;  // DCO+ set,
                                 // freq = xtal x D x N+1
SCFI0 |= FN_4;                   // x2 DCO freq,
                                 // 8MHz nominal DCO
SCFQCTL = 121;          // (121+1) x 32768 x 2 = 7.99 MHz
```

- Reference voltage configuration:

```
ADC12CTL0 = REF2_5V | REFON;      // Internal 2.5V ref on
```

- DAC12 configuration:

```
DAC12_0DAT = 0x00;              // DAC_0 output 0V
DAC12_0CTL = DAC12IR | DAC12AMP_5 | DAC12ENC;
                                // DAC_0 -> P6.6,
                                // DAC_1 -> P6.7,
                                // DAC reference Vref,
                                // 12 bits resolution,
                                // Immediate load,
                                // DAC full scale output,
                                // Medium speed/current,
                                // Straight binary,
                                // Not grouped
```

- Timer_A configuration:

```
// Before entering in LPM0:
TACTL = TACLR | MC_1 | TASSEL_2;        // up mode, SMCLK
// Timer_A ISR:
TAR = 0;                  // TAR reset
TACCR0 = 13600;           // Delay to allow Ref to settle
TACCTL0 |= CCIE;          // Compare-mode interrupt
TACTL = TACLR | MC_1 | TASSEL_2;        // up mode, SMCLK
```

- Configuration of ports:

```
// SW1 and SW2 ports configuration
P1SEL &= ~0x03;           // P1.0 and P1.1 I/O ports
P1DIR &= ~0x03;           // P1.0 and P1.1 digital inputs
P1IFG = 0x00;             // clear all interrupts pending
P1IE |= 0x03;             // enable port interrupts

// P6.6 (DAC12_0 output)
// There is no need to configure P6.6 as a
// special function output since it was configured in the
// DAC12 configuration register (DAC12_0CTL) using
// DAC12OPS = 0
```

10.5 Quiz

1. The MSP430 DAC12 module has which architecture?

(a) Binary Weighted DAC;

(b) Interpolating DAC;

(c) Thermometer coded DAC;

(d) R/2R Ladder DAC.

2. In an R/2R Ladder DAC architecture the equivalent resistance between V_{REF} and ground is:

(a) R/2;

(b) R;

(c) 2R;

(d) 4R.

3. For a DAC with a Differential NonLinearity (DNL) less than 1 LSB:

(a) The transfer function deviates from a straight line;

(b) The output voltage value is 0 when the digital input is 0;

(c) The full-scale output voltage is equal to maximum digital input;

(d) No data is lost.

4. Filtering is important to DAC operation because it:

(a) Increases resolution;

(b) Reproduces a signal precisely up to the Nyquist frequency;

(c) Can provide an approximate smooth continuous time signal;

(d) Spreads noise over more frequencies.

5. To provide a DAC12's analogue output voltage corresponding to three times the reference voltage with 12-bit resolution:

(a) DAC12RES = 0 and DAC12IR = 1;

(b) DAC12RES = 0 and DAC12DF = 1;

(c) DAC12IR = 1 and DAC12DF = 0;

(d) DAC12RES = 1 and DAC12IR = 1.

6. To update the DAC12's analogue output voltage with a rising edge of the Timer_A CCR1 output:

(a) DAC12LSELx = 3;

(b) DAC12LSELx = 2;

(c) DAC12LSELx = 1;

(d) DAC12LSELx = 0.

Solution: 1. (d); 2. (b); 3. (d); 4. (c); 5. (a); 6. (b)

10.6 FAQs

1. Why does the DAC12 analogue output voltage produce sharp variations while using the DMA controller to transfer data?

One of the reasons may be due to the DAC12 output voltage settling time, because the DMA controller can transfer data faster than the DAC12 output can settle. The update of the analogue output voltage may cause these sharp changes.